MONITORING TRUTH & FLEET RECONCILIATION SERIES · EC-WP-1103

A Rules Engine That Measures Its Own Usefulness

NORA: the not-needed rate, four rules that could never fire, and a rule that fired on the healthy sites.

Every alarm system grows monotonically, because nobody removes a rule. This is the measurement that breaks the ratchet: a rules engine instrumented to keep score of how often its own output was ignored. Included is an honest account of what the same instrumentation found underneath it. Most of the defects in this reference are ours.

energycomplianceinc.com **OCTOBER 2026**

Disclaimer & Legal Notice

This publication is the work product of Energy Compliance, Inc. and is provided for general information and educational purposes only. It does not constitute legal advice, regulatory guidance, engineering advice, or formal interpretation of any NERC Reliability Standard, FERC order, Department of Energy directive, RTO or ISO protocol, or state commission rule.

This publication may not be used or cited as authority in any audit, enforcement, regulatory, or legal proceeding.

On the absence of regulatory citations

This series carries no requirement text, no requirement number, no standard version, and no effective date, and it cites no order, protocol section, or rule. The omission is deliberate. The subject of the series is how a monitoring platform represents measured reality. That question comes before, and stands apart from, which requirement is being satisfied. Regulatory standard families are named only where the connection to a specific finding is material, and they are named as families, never as citations. Nothing here should be read as a statement of what any standard requires. A reader who needs operative language must obtain it from the current enforceable source for the relevant registered entity and audit period.

On the findings

Every defect described here was measured against a live production monitoring platform. The observations are anonymized: no client, site, site code, counterparty, vendor, product, asset tag, or person is identified, and figures are reported at a level of aggregation that does not permit re-identification. Several of the defects described are in RenewOps, the platform Energy Compliance builds. Those are marked as ours and written up on the same terms as the rest.

On currency

A measurement is a statement about one system at one moment. The platforms described are under active remediation. A figure that was true when measured may not describe the same system today, and it does not describe any other system at all. Where a matter was open, unresolved, or unverified at the time of writing, it is identified as such. Energy Compliance, Inc. does not represent that any condition described here persists, or that any condition absent here does not exist elsewhere.

This edition was reviewed and corrected on October 17, 2026. The measurements were not re-taken. They remain the August 24–26, 2026 figures, and status statements about the platform are dated to that window.

Non-affiliation

NERC, FERC, the United States Department of Energy, the Public Utility Commission of Texas, ERCOT, PJM, and the names of Regional Entities are referenced here for descriptive purposes only. Energy Compliance, Inc. is not affiliated with, sponsored by, or endorsed by the North American Electric Reliability Corporation, the Federal Energy Regulatory Commission, the Department of Energy, the Public Utility Commission of Texas, ERCOT, PJM Interconnection, L.L.C., or any NERC Regional Entity. RenewOps is a product of Energy Compliance, Inc.

Energy Compliance, Inc. assumes no responsibility or liability for errors, omissions, or outcomes resulting from the use of this publication. Compliance obligations depend on facts and circumstances unique to each registered entity, and the reader should consult qualified counsel and subject-matter experts when making compliance decisions.

Foreword

This professional reference is one of a series Energy Compliance, Inc. publishes for registered entities and the people who run their compliance programs.

I have spent more than thirty years on every side of the bulk electric system. I have operated control centers as a Reliability Coordinator, Transmission Operator, and Power System Operator. I have audited grid facilities and signed off on findings as a senior compliance auditor. I have worked enforcement matters from inside the regulator's process. For the last several years I have advised registered entities directly through the firm I founded.

Here is the thing about a screen. In a control room the screen is not a report on the system. For the eight hours you sit in front of it, the screen *is* the system. You act on it. You hand over on it. You write it into the log, the log becomes the record, and the record becomes the evidence. Everything downstream inherits whatever the screen got wrong, and nothing downstream is equipped to notice.

That is the subject of this series. Not whether a fleet is performing. Whether the number on the screen is a measurement of the fleet at all.

The series came out of an assessment. In 2026 we pointed a defect battery at a live production monitoring stack running a multi-gigawatt utility-scale PV and storage fleet and wrote up everything it found: twenty-plus verified defects, each with a source and a timestamp. What I did not expect was the shape of the result. The defects were not a

list of unrelated bugs. They were six recurring structures, and every one of them was a system reporting an absence as a value. A missing irradiance reading rendered as zero. A denominator that shrank as equipment failed, so a half-dead site scored a perfect hundred. An ingest timestamp wearing an event timestamp's label. A reading four hundred days old displayed as current. A revenue meter an order of magnitude above nameplate rendered live, because nothing in the system had a nameplate to check it against. A voltage band seeded from a plant's own history, indistinguishable on screen from a schedule issued by the entity of record.

None of that was a vendor defect, and I want to be careful here, because the easy version of this series is a hit piece on monitoring platforms and that version would be wrong. A monitoring platform displays what it is given. It was built to display. It was never asked to reconcile what it displays against what a contract requires, what a standard demands, or what physics permits. Nobody built that layer. The gap is not a failure of any product. It is a category that does not yet exist in most fleets.

Two of the defects in the original assessment were in our own platform. We wrote those up the same way and shipped the report with them in it, and they appear in these references marked as ours. I mention it because it is the only credential in this series that is hard to fake. Any vendor can show you a findings list from someone else's product. A vendor who has never found a fault in their own has not looked.

These references carry no regulatory citations, and that is deliberate; the reasoning is on the previous page. They are also not marketing material dressed as content. I write them in the same voice I use in a control room or in front of a Regional Entity audit team. Direct. Grounded in evidence. Honest about what a measurement can and cannot tell you.

The not-needed rate is the measurement I am proudest of and the one that has cost us the most. It is a rules engine keeping score of how often its own output was ignored, which is an uncomfortable number to publish and an easy one to quietly stop computing. Most of the defects in this reference are ours. Four rules that could never fire under any input. One rule firing on the healthy sites. We found them because the engine was built to find them.

Energy Compliance, Inc. exists because most of the consulting offered to registered entities is structured for billable hours rather than for outcomes. A senior practitioner oversees every engagement, and we do not bring five people to a meeting that needs one. We automate the work that should be automated and apply senior judgment to the work that requires it. If that matches what you are looking for, the back of this reference tells you how to reach us.

If not, the reference still belongs to you. Take what is useful. Apply it well. And keep asking the only question that matters about a number on a screen: what would have to be true for this to be wrong, and would I be able to tell?

Rob Smith

Founder and Principal, Energy Compliance, Inc.

Bloomington, Minnesota

How to Use This Reference

This reference is one of eight in the Energy Compliance Monitoring Truth and Fleet Reconciliation Series. The series is organized layer by layer. Each reference stands on its own and assumes none of the others, at the cost of a small amount of repeated framing where two layers meet.

EC-WP-1100 · The Measurement Layer: Missing Is Not Zero
The six-class failure taxonomy and the five design rules that answer it. This is the foundational reference for the series; every paper that follows takes one class apart on a live system.

EC-WP-1101 · The Alarm Layer: The Board That Deleted Its Own Alarms
Alarm management treated as an information system: load, priority, latching, and the monitor that stops. Seven failure modes, each with the arithmetic that exposes it.

EC-WP-1102 · The Operator Layer: What the Incoming Operator Cannot See
Situational awareness as a testable property of the record rather than a quality of a dashboard. Turnover, attribution, and identity, measured five ways.

EC-WP-1103 · The Rules Layer: A Rules Engine That Measures Its Own Usefulness (this reference)
The not-needed rate: how a rules engine earns the right to keep a rule. It is the measurement that breaks the monotonic-growth ratchet in an alarm catalog.

EC-WP-1104 · The Procedure Layer: A Machine May Draft It

Switching order integrity when a machine prepares the draft. Five integrity requirements, each with the failure that motivates it, and the one field a drafter must never write.

EC-WP-1105 · The Contract Layer: Six Ways to Compute 99%

Availability constructs reconciled against executed agreements. Computability as a per-site position (ready, partial, blocked, or undefined) with the blocker named.

EC-WP-1106 · The Evidence Layer: Displayed Is Not Citable

Provenance as the property that separates a number you can show from a number you can cite. The tautology trap, the seeded schedule, and the obligation nobody could mark done.

EC-WP-1107 · The Platform: The Reconciliation Layer

RenewOps itself. What the platform does, what it deliberately refuses to produce, and what it does not claim. This is the capability reference for the layer the other seven show the need for.

How to read a figure in this series

Read every number with its source, and with its measurement date where the text gives one. This series does not make claims about systems it has not measured. Where a value could not be verified, it is marked as deferred with the specific question that would resolve it, rather than softened into an estimate.

What these tests cannot tell you

Every diagnostic in this series is a test of the *record*, not of the plant. A clean result means the platform's representation of the fleet is internally consistent, sourced, and current. It does not mean the equipment is healthy, the revenue is correct, or the program is compliant. The reverse also holds: a failed test is evidence that a number

cannot be defended, not evidence that the underlying asset is in trouble. Several of the worst findings in the original assessment sat on top of plants that were running perfectly well. Conflating the two is the most common misreading of this work, and it produces false alarm and false comfort in equal measure.

Specific to this reference: a not-needed rate tells you whether a rule is earning its place on the board. It cannot tell you whether a rule that never fires is worthless or is guarding against something that has not happened yet. That distinction is a judgment about consequence, not a statistic about frequency, and the reference is explicit about where the arithmetic stops and the judgment starts.

About this document

NORA is the operator response layer inside RenewOps, the platform Energy Compliance builds. This document describes how it works, what it got wrong, and what those mistakes taught.

That framing is deliberate. **Most of the defects in this paper are ours.** A rules engine is easy to describe well and hard to build honestly, and the interesting content is entirely in the second part.

All findings are anonymized. No client, site, site code, vendor, asset tag or person is identified. Regulatory standard families are named only where the connection is material; no requirement text, requirement number, version or effective date appears here.

The findings in this reference are drawn from an assessment of one fleet. They are offered as a taxonomy to test your own fleet against, not as a statement about it.

Executive Summary

A rule that cannot be retired is not a rule

Every alarm system we have assessed grows monotonically. Nobody removes a rule, because removing one means admitting somebody added a bad one. So the flood worsens every year and operators stop reading the board.

NORA is built around one measurement intended to break that ratchet: **every time a rule surfaces something, the operator records whether it was worth surfacing.** That proportion is the not-needed rate, and it sits in the rule list where everyone can see it. A rule operators keep dismissing gets retired, not defended.

The mechanism is not technically difficult. It works because the number is published, per rule, and because retirement is the expected outcome rather than a failure.

Rule	Operator answers	Not-needed rate
Site communications loss	66	0–1.5%
Alarm flood	21	0–1.5%
Stale data feed	13	0–1.5%
Protection element standing long	12	0–1.5%
Voltage excursion, 20-minute companion	7	0–1.5%
Compliance preparation window	6	0–1.5%

Six enabled rules, every one under the retirement line. That is the headline, and taken alone it would be a marketing document. The rest of this paper is what the same assessment found underneath it.

Four of seven rules were structurally incapable of firing. The health page read "quiet" and "no independent check available" while 69,590 alarms crossed the fleet in twenty-four hours.

Defect	What it meant
Four rules could never fire	An inner join to a signal producer that did not exist silently dropped them from the candidate set. They appeared enabled and healthy.
The communications rule was inverted	It counted communications-loss alarms, which a dark site cannot send. It fired on healthy sites and stayed silent on the four that were dark.
Quiet periods keyed per site	All 28 protection conditions muted for a week after one operator answered eleven prompts in ninety-nine seconds.
A threshold set to zero	An availability rule with its threshold at 0% cannot be exceeded.

Each of those is the same failure this practice keeps finding, in a new costume: **a system that could not distinguish "nothing is wrong" from "nothing is reaching me."** A rule that never fires and a rule with nothing to report produce identical output.

1 What a rule has to earn

A rule occupies an operator's attention, which is the scarcest resource in a control room. It has to pay rent, and the rent has to be measured.

1.1 The four properties

Property	Means	Measured by
Actionable	There is something the operator receiving it is authorized to do.	Not-needed rate
Evidenced	The system can actually produce the signal it fires on, under the conditions where the rule matters.	Producer coverage (Ch. 3)
Correctly polarized	Its evidence exists when the condition is true, not only when it is false.	Inversion test (Ch. 4)
Bounded	It cannot flood, and its quiet period is scoped to what the operator actually dismissed.	Prompt rate (Ch. 5)

Most rule engines test the first property and assume the other three. The assessment described here failed all three assumed ones and passed convincingly on the one that was measured.

1.2 The rule record

A rule in NORA is a row, and the fields it carries are the argument of this chapter. What a rule has to declare about itself:

- **signal**: constrained vocabulary; see Chapter 7
- **threshold**: refused below a floor; see below
- **evaluation_interval**: refused under five minutes
- **quiet_period**: and the KEY it is scoped to; see Chapter 5
- **prompt_text**: what the operator is actually asked
- **enabled**: and enabled is not the same as capable
- **not_needed_rate**: published, per rule, visible to everyone

Two of those carry refusals rather than defaults, and both refusals exist because of observed failures. **An evaluation interval under five minutes is refused.** A rule that evaluates faster than an operator can respond is a flood generator with a threshold. And **a rule change with no reason is refused**, because a threshold that moved with no recorded rationale cannot be reviewed later, and thresholds drift toward whatever silences the complaints.

The distinction the rest of this paper turns on. *Enabled* means somebody switched the rule on. It does not mean the rule can fire. Four of the seven rules assessed were enabled, healthy-looking, and structurally incapable of producing a single prompt. Nothing in the interface distinguished them from rules that simply had nothing to report.

2 The not-needed rate

This is the measurement that lets an alarm set shrink. It is not a technical innovation. It is a willingness to record an answer nobody wants to hear.

2.1 The mechanic

When a rule surfaces something, the operator is asked to close it with one of a small set of outcomes. One of those outcomes is *this did not need surfacing*. The proportion of closures carrying that answer is the rule's not-needed rate, and it is published in the rule list beside the rule's name.

Calculation 1 · Not-needed rate, with a confidence gate

```
SELECT r.rule_key, r.enabled,
COUNT(a.*) AS answers,
COUNT(*) FILTER (WHERE a.outcome = 'not_needed') AS not_needed,
ROUND(100.0 * COUNT(*) FILTER (WHERE a.outcome = 'not_needed') / NULLIF(COUNT(a.*), 0),
1) AS not_needed_pct,
-- do not retire a rule on three answers
(COUNT(a.*) >= 20) AS sample_sufficient,
CASE WHEN COUNT(a.*) < 20 THEN 'observe'
WHEN COUNT(*) FILTER (WHERE a.outcome = 'not_needed') * 100.0 / COUNT(a.*) >= 40 THEN
'RETIRE'
WHEN COUNT(*) FILTER (WHERE a.outcome = 'not_needed') * 100.0 / COUNT(a.*) >= 15 THEN
'retune'
ELSE 'earning its place' END AS verdict
FROM nora_rule r
LEFT JOIN nora_answer a ON a.rule_key = r.rule_key
GROUP BY r.rule_key, r.enabled
ORDER BY not_needed_pct DESC NULLS LAST;
```

2.2 Reading the verdict

Verdict	Condition	Action
observe	Fewer than 20 answers	Insufficient sample. Do not act, and do not publish the rate either.
earning its place	Under 15% not-needed	Leave it alone.

Verdict	Condition	Action
retune	15–40%	Threshold or scope is wrong. Fix the rule before retiring it.
RETIRE	40% or above	Two of every five prompts wasted an operator's attention. Retire it.

The observed rules all returned 0–1.5%, well inside "earning its place." But four of the six had fewer than twenty answers, which is exactly the case the sample gate exists for. A rate computed from six answers is not evidence that a rule is good. It is evidence that a rule has fired six times.

The trap this measurement sets for itself. A rule that cannot fire has no answers, so it has no not-needed rate, so it never appears in the retirement queue. Its silence reads as virtue. Four of the seven rules assessed were sitting in exactly that position, and the not-needed rate on its own would have protected them indefinitely. That is Chapter 3.

3 Four rules that could never fire

The candidate view assembled signals from four producers and inner-joined them to the rule table. Any rule whose signal had no producer was silently dropped from the result. It stayed enabled and visible, and it could not fire.

3.1 The mechanism

Simplified. The join is the defect.

```
SELECT r.rule_key, s.site_id, s.value
FROM ( SELECT 'comms_condition' AS signal, ... FROM ...
UNION SELECT 'alarm_flood' AS signal, ... FROM ...
UNION SELECT 'protection_long' AS signal, ... FROM ...
UNION SELECT 'compliance_due' AS signal, ... FROM ... ) s
INNER JOIN nora_prompt_rule r ON r.signal = s.signal; -- here
-- Rules whose signal is not in the union simply vanish.
-- No error. No warning. No row in any diagnostic.
```

Four signals had rules but no producer: alarm rate, stale data, availability below guarantee, and irradiance reference. Each rule existed, was enabled, and had a threshold. The inner join removed them from consideration on every evaluation cycle.

3.2 What that concealed

Because the rules could not fire, the health page reported *"quiet"* with the note *"no independent check available."* At the same moment, measured directly:

Actual condition	Measured
Alarms across the fleet in 24 hours	69,590
Implied hourly rate against a 60/hr threshold	2,900/hr, 48 times the threshold
Individual sites independently over threshold	8
Dead communications links	4
Availability rule threshold as configured	0%

The last row deserves its own note. A threshold of zero cannot be exceeded. Even had the producer existed and the join succeeded, that rule would still never have fired. That is a second, independent reason for the same silence, hidden behind the first.

Calculation 2 · Producer coverage. Run this on any rule engine.

```
-- Every rule whose signal no producer emits. Should return zero rows.
SELECT r.rule_key, r.signal, r.enabled, r.threshold, 'NO PRODUCER' AS defect
FROM nora_prompt_rule r
WHERE NOT EXISTS (SELECT 1 FROM nora_signal_producer p WHERE p.signal = r.signal)
UNION ALL
-- Every enabled rule that has produced nothing in the last thirty days.
SELECT r.rule_key, r.signal, r.enabled, r.threshold, 'ENABLED BUT SILENT' AS defect
FROM nora_prompt_rule r
WHERE r.enabled
AND NOT EXISTS (SELECT 1 FROM nora_prompt pr WHERE pr.rule_key = r.rule_key AND
pr.created_at > now() - interval '30 days');
```

The second half of that query is the more general control. **An enabled rule that has produced nothing in thirty days is making a claim**: that its condition has not occurred. That claim deserves a check rather than an assumption.

3.3 The design rule

A rule declares its producer, and the engine refuses a rule whose producer does not exist.

Not a warning, not a log line. A refusal at write time, in the same function that already refuses an interval under five minutes. A rule that cannot fire should be impossible to save, because once saved it is indistinguishable from a rule with nothing to report.

And **the health surface must separate three states**: *evaluated, no condition found*; *not evaluated, producer missing*; and *not evaluated, evaluator down*. Collapsing those into "quiet" is the same substitution failure that runs through this entire body of work. It renders an absence as a reassuring value.

4 The rule that fired on the healthy sites

A communications-loss rule counted communications-loss alarms. A site that has lost communications cannot send one.

4.1 The inversion

The rule's evidence was the arrival of alarms of a particular type from a site. Its premise was that more comms alarms mean worse communications. That premise is reasonable, and over the middle of the range it is true.

At the boundary it inverts completely. A site whose link has failed entirely sends nothing, including comms alarms. Its count is zero. Zero is the healthiest possible value under the rule's logic.

Observed behavior: the rule fired on four sites whose links were reported healthy, and stayed silent on the four sites that were completely dark.

The rule was not merely wrong. It was precisely, systematically anti-correlated with the condition it existed to detect.

4.2 The general form

This is worth generalizing, because it is a whole class rather than one bug.

The inversion test. For any rule, ask: *if the condition were completely true, would the evidence still arrive?*

If the thing that is failing generates the evidence, the answer is no, and the rule inverts at exactly the point where it matters most. Rules built on *counts of incoming events* are the usual carriers. Rules built on *absence of expected events* are usually safe.

Rule built on	Behavior at total failure	Safe?
Count of comms-loss alarms	Goes to zero. Reads as perfect health.	No
Time since last row from site	Grows without bound. Fires.	Yes
Count of error responses	Goes to zero when the service is down entirely.	No
Time since last successful poll	Grows. Fires.	Yes
Ratio of bad readings to total	Undefined when total is zero.	No

The repair was to replace the rule with a stale-feed rule that reads a link-health view separating *rows arriving* from *values arriving* (Chapter 9). That distinction survives total failure because it is built on expected arrivals rather than observed ones.

4.3 Why it survived review

Because it fired. A rule that produces prompts looks alive, and the prompts it produced were not obviously wrong; the sites it named did have communications events. Nobody compares the set of sites a rule fires on against the set of sites that are actually failing, because doing so requires an independent source of truth about failure. That is the thing the rule was supposed to be.

5 Quiet periods, and what an operator actually dismissed

One operator answered eleven prompts in ninety-nine seconds. The quiet period that followed muted all twenty-eight protection conditions across the fleet for a week.

5.1 What happened

Eleven prompts in ninety-nine seconds is nine seconds per prompt. That is not consideration. It is clearing a queue. The interface asked eleven similar questions in succession and the operator did the rational thing.

The quiet period then did what it was configured to do and suppressed further prompts. But it was **keyed per site rather than per condition**, so dismissing eleven prompts about specific protection elements silenced every protection condition at those sites, including conditions the operator had never been shown.

Setting	As configured	Consequence
Quiet period key	**site**	Dismissing one condition mutes all conditions at that site
Quiet period duration	7 days	A nine-second decision buys a week of silence
Protection conditions muted	28 of 28	The entire category, fleet-wide

5.2 The two design errors, separately

Scope. A dismissal is a statement about *the thing dismissed*. It says nothing about a different condition on the same asset, and considerably less than nothing about a different asset. The quiet period must be keyed to the narrowest identifier that describes what the operator actually saw. That means condition and asset, not category and site.

Duration versus deliberation. Seven days is a reasonable quiet period for a considered dismissal and an unreasonable one for a nine-second one. The system had the information to tell them apart, because it knows how long the prompt was on screen. It did not use it.

Calculation 3 · Quiet period scaled to deliberation

```
-- Detect queue-clearing, and let the quiet period reflect it.
WITH answered AS (
SELECT actor_code, rule_key, prompt_id, answered_at,
EXTRACT(epoch FROM (answered_at - shown_at)) AS seconds_on_screen,
COUNT(*) OVER (PARTITION BY actor_code ORDER BY answered_at RANGE BETWEEN interval '5 minutes' PRECEDING AND CURRENT ROW) AS answers_in_5min
FROM nora_prompt_answer )
SELECT actor_code, prompt_id, seconds_on_screen, answers_in_5min,
CASE WHEN seconds_on_screen < 15 OR answers_in_5min > 5
THEN interval '4 hours' -- queue-clearing: short mute
ELSE interval '7 days' -- considered: full quiet period
END AS quiet_period,
(seconds_on_screen < 15 OR answers_in_5min > 5) AS rapid_dismissal
FROM answered;
```

The rapid_dismissal flag is worth publishing on its own, apart from what it does to the quiet period. A rule accumulating rapid dismissals is telling you something the not-needed rate will not: operators are not evaluating it. Its not-needed rate is then measuring the interface rather than the rule.

The design rule. A dismissal binds only what was dismissed, and for a period proportionate to the attention it received.

Key the quiet period to condition plus asset. Scale the duration by observed deliberation. Publish the rapid-dismissal rate per rule beside the not-needed rate and read the two together. A rule with a low not-needed rate and a high rapid-dismissal rate is not earning its place. It is being waved through.

5.3 The prompt itself is part of the rule

Eleven near-identical prompts in a row is an interface failure, not an operator failure. Where a condition affects many assets at once, the correct object is *one prompt describing the group*, with the members listed inside it. One decision, correctly scoped, recorded once against everything it covers.

6 The procedure library: retrieval that cites

When a rule surfaces something, the response draws on the entity's own controlled procedures, cited to document and section. The operator reads their procedure, not a vendor's idea of one.

6.1 What was loaded

Measure	Value
Controlled documents ingested	121 of 121
Chunks embedded	1,203
Chunk size / overlap	600 tokens / 80
Documents classified as procedures	115
Alarm response guides	4
Switching practices	2
Documents carrying a standard linkage	20
Reconciliation after load	0 to load, 121 already done

6.2 The classification decision, and why it mattered

None of the ingested documents is a reliability standard. They are the entity's own procedures, which *implement* standards. That distinction determines what the retrieval layer is allowed to answer with.

An earlier load let automatic detection read internal document numbers as standard numbers, and it filed eight procedures as standards. Had that stood, an operator asking what a particular standard requires would have received the entity's own procedure instead of the standard text already in the index.

That is a provenance failure in the sense of EC-WP-1100, where a document of one epistemic status is rendered as another, and in a compliance context it is the expensive kind. A procedure describes what an entity chose to do. A standard describes what is required. Confuse them in an evidence conversation and the entity ends up citing itself.

Two live findings the load surfaced in the documents themselves

Ninety-nine of the 121 still carried a template document number rather than a real one. An unnumbered controlled document is an audit finding on its own, and it was raised ahead of the library's effective date rather than discovered later.

The library was future-dated. Every document carried an effective date ahead of the load. The library existed and was not yet in force. The retrieval layer must carry that distinction, because answering an operator today from a procedure that takes effect next month is wrong in a way that looks entirely right.

6.3 The ingest path, and why it is audited

Documents enter through a function that **re-verifies the operator's role against the roster on every call** and writes an audit row per ingest. That is deliberate. A procedure library is an evidence source, and an evidence source with an unaudited write path is not one.

Supersession is explicit. A replacing document names the document it retires, and the retired version stays queryable with a retired status rather than being deleted. A procedure library that cannot show what was in force on a past date cannot support an event reconstruction, and event reconstruction is most of what it exists for.

6.4 Three defects in our own tooling, and the shape they shared

Defect	Consequence
A failed listing returned an empty array	The supersession map came back empty, so the loader began writing second copies of documents alongside the ones they replaced. Six landed before it was stopped.
Three concurrent workers	Throughput did not change, because the function serializes anyway. Per-request latency tripled past the gateway timeout, and the extra streams starved the connection pool the control room reads through.
Only one exception type caught	A transport-level disconnection 44 documents into the run killed it silently.

All three are the same shape: a failure that looked like a result. An empty list read as "nothing to do." A timeout read as an answer. A dropped connection read as the end of the work.

That sentence is the thesis of this entire body of work, arrived at from a completely different direction. The fixes were correspondingly simple. Retry and then *abort* rather than trusting an empty answer. Run one worker. Catch transport failures and retry them, so that only a real answer from the function, good or bad, ends the loop.

7 What the vocabulary permits

The signal vocabulary is a constraint with exactly nine permitted values. Widening it is a governance decision about what the system is allowed to assert, not a repair.

7.1 Why constrain the vocabulary at all

With an unconstrained signal field, every rule can invent its own vocabulary, producers and consumers drift apart silently, and the inner-join failure of Chapter 3 becomes undiagnosable. The constraint is what makes producer coverage checkable. It has a second effect, less obvious and more important: **it forces the question of what the system is entitled to claim.** Each permitted signal is a category of assertion the platform is prepared to make to an operator.

7.2 The signal that is missing

None of the nine permitted signals describes this: *the value arrived on time, from a healthy link, and is not physically possible.*

That gap is why the plausibility failures in EC-WP-1100 could not raise an operator prompt. A meter reading orders of magnitude above nameplate, mixed units in a single column, a setpoint of a different order from the quantity it regulates: the rules engine had no vocabulary for the claim. The proposed shape is straightforward: a plausibility signal, a per-site band table sourced from voltage class and nameplate, an audit row per refusal, and the same effectiveness measurement as the other rules.

> **It was proposed and deliberately not added**
>
> As of the August 2026 measurement, the signal had not been added. Widening the constraint governs what NORA is allowed to say to an operator. That is a platform decision with consequences beyond one rule, and it belongs to the person accountable for the platform rather than to whoever happened to be in the codebase that night. Recording it as a named, pending decision is the same discipline as the held catalog decisions in EC-WP-1101.

8 Ten questions for your own rules engine

#	Question	What a bad answer means
1	Does every enabled rule have a producer that actually emits its signal?	Any that do not are enabled, invisible and incapable. Run this first.
2	Which enabled rules have produced nothing in thirty days?	Each is asserting its condition has not occurred. Verify rather than assume.
3	For each rule: if the condition were completely true, would the evidence still arrive?	If not, the rule inverts precisely where it matters.
4	Does any rule have a threshold that cannot be exceeded?	A threshold of zero, or one above the physical maximum, is a disabled rule wearing an enabled badge.
5	Do you publish a not-needed rate per rule?	Without it your rule set can only grow.
6	Is the not-needed rate gated on sample size?	A rate from six answers is not evidence a rule is good.
7	What is a quiet period keyed to?	Site-keyed means dismissing one condition mutes conditions the operator never saw.
8	How long was the prompt on screen before it was answered?	Under fifteen seconds is queue-clearing, and it should not buy a week of silence.
9	Does your health surface distinguish "no condition" from "not evaluated"?	If both render as "quiet," quiet means nothing.
10	Can your response layer cite the document and section it answered from?	An uncited answer is a vendor's opinion presented as your procedure.

Every alarm system grows monotonically until somebody measures whether the rules are worth having. The measurement is not hard. Publishing it is.

9 What RenewOps does about this

The reconciliation layer. It reads your monitoring stack and checks it against ratings, agreements and the documents of record. It declines to publish what it cannot defend.

NORA is the rules engine inside RenewOps, and the subject of this reference. What follows is what it does, what it refuses to do, and the four things it got wrong that its own instrumentation caught.

9.1 The measurement that earns a rule its place

Every rule is scored by its **not-needed rate**, the share of its prompts an operator answered "not needed." On 25 August 2026 six enabled rules all measured between zero and 1.5 percent, with answer counts of 66, 21, 13, 12, 7 and 6. Each sat under the retirement line on the number, not on anyone's opinion. Four of the six, though, were below the twenty-answer gate in Chapter 2, so by this reference's own rule those four rates are still observations rather than verdicts.

A rule has three legible states, not two. **Enabled. Disabled, with the reason written onto the rule record.** And **NOT WIRED**: a rule that exists but has no candidate branch and therefore could never fire under any input. That third state is the one this reference argues matters most. From outside, a rule with no candidate branch and a rule that is quiet because the fleet is healthy look identical. They mean opposite things.

How that was found. On 24 August 2026 four of seven rules structurally could not fire, while the screen printed *quiet* and *no independent check available* for each of them. An operations manager reading that screen would reasonably have concluded the fleet was clean on four dimensions it was not being checked on at all. Enabled rules went from six to nine on 25 August, and from that date the wiring state was rendered separately from the firing state.

9.2 The inverted rule

The communications rule counted comms-loss alarms. It fired on healthy sites and stayed silent on dark ones, for a reason that reads as obvious once stated: **a site that has gone dark cannot send you an alarm saying it has gone dark.** The rule was measuring the presence of a message from a site, which is close to the opposite of the condition it was named for.

It was replaced by a stale-feed rule that asks the inverse question, how long since this feed last said anything, which covers dark sites by construction. On enable it flagged twelve of thirty feeds stale, from 796 days down to 3.9 days, per feed and per site.

9.3 Two gates every candidate branch has to pass

A rule comparing a *partial* day's measurement against a *full* day's model made the entire fleet look faulted every morning and recover every afternoon. Seventeen sites read suspect on one day and normal on the next, with nothing having changed at any plant.

Two requirements came out of that, and they apply to any candidate branch:

1. **A completed-day gate.** A comparison against a daily model does not run until the day it describes has finished.
2. **A persistence test.** A condition that appears once and clears is not a finding. The genuinely faulted references are the persistent ones, and the rule is written to say so.

Re-enabled on completed days, the rule caught four genuine site-days in seven: sites generating between 1,000 and 1,800 MWh against **zero reported sunlight**. The clearest was 1,059.6 MWh generated against zero reported sunlight. That is the finding the morning artifact had been burying.

9.4 What the rules caught once they could fire

After the producers were built and the rules repointed:

Rule	What it caught on enable
Irradiance reference suspect	Four completed site-days in seven, as above.
Stale data feed	**Twelve of thirty feeds stale.** Longest 796 days; then 424, 193, 104, 82, 34, 29.5, 27.5, 13, 11, 3.9.
Voltage notification window	Zero open, and *correctly* so. All state changes on record were historical.

The third row is the one to dwell on. A rule that returns zero *after* its producer exists and its logic is verified is a meaningful zero. Before that, the same zero was meaningless. The output is identical; only the epistemic status changed.

9.5 The rules it declines to enable

Rule	Why it is off
Availability below guarantee	Deliberately disabled, correctly, until the capability score is persisted server-side. A rule that fires off a number computed in a browser session produces a prompt nobody can audit afterwards.
Alarm flood	The only available count does not mean what the rule needs it to mean. Pointing the rule at it would have put a fabricated alarms-per-hour figure in front of an operator, so the signal is left unconsumed rather than approximated.
Plausibility	The permitted signal vocabulary contains no signal for *arrived on time, from a healthy link, and is not physically possible*. Widening that vocabulary governs what NORA is allowed to say at all, so it is treated as a platform decision for the owner rather than a repair an engineer makes.

The configuration it refuses. The rule-writing path rejects settings that would make a rule useless; an evaluation interval under five minutes is refused outright. The engine guards its own content. It does not guard the identity of whoever changed a rule, which is a caller-supplied string, and that is stated below rather than glossed.

9.6 The quiet period, and what it cost

RenewOps applies a per-site quiet period to suppress repeat prompts for the same condition. It worked, and then it over-worked: one user answering eleven prompts in ninety-nine seconds muted twenty-eight standing protection conditions at once. As of the August 2026 measurement the quiet period had been shortened. Scoping it per condition rather than per site was owed and not yet done.

This is the failure mode a not-needed rate is blind to. A rule that never prompts scores perfectly on a measure of whether its prompts were useful.

9.7 What the instrumentation cannot tell you

The not-needed rate is a real measurement taken over an attribution trail that is **not authenticated**. The client supplies the responder's name. The measurement is sound as a count of how often prompts were dismissed. It is not evidence of *who* dismissed them.

And as of the August 2026 measurement, no rule had been retired on the measure. The lifecycle this reference can evidence runs from enabled, to disabled with a reason, to not wired. Retirement is the step the not-needed rate exists to justify, and it had not been exercised, because on the fleet measured no enabled rule had crossed the retirement line. A ratchet that has never had to release is not yet proven to release.

A rules engine that publishes how often it was ignored is making an argument against its own headcount. That is the point. The engines that grow without limit are the ones that never had to make it.

Figures in this chapter measured 24–26 August 2026 on a live production instance. They describe that instance on those dates and are not a claim about any other fleet.

Glossary of Terms

Term	Meaning in this paper
Candidate branch	The part of the rules engine (the candidate view in Chapter 3) that assembles the signals a rule is evaluated against. Without one, a rule cannot fire.
Completed-day gate	A requirement that a comparison against a daily model does not run until the day it describes has finished.
Inner join	A database join that returns only rows with a match on both sides. In Chapter 3 it silently dropped every rule whose signal had no producer.
Inversion test	Whether a rule's evidence would still arrive if its condition were completely true. A rule that fails it goes silent at total failure.
NORA	The operator response layer and rules engine inside RenewOps.
NOT WIRED	The rule state in which a rule exists but has no candidate branch, so it can never fire.
Not-needed rate	The share of a rule's prompts that operators closed as *this did not need surfacing*, published per rule.
Persistence test	A requirement that a condition persist before it counts as a finding; a condition that appears once and clears is not one.
Producer	The process that emits a signal a rule depends on. Producer coverage is the check that every enabled rule's signal has one.
Prompt	A question NORA puts to an operator when a rule's condition is met.
Provenance	The recorded origin and epistemic status of a value or document: measured, modeled, inferred, or issued.
Quiet period	The interval after a dismissal during which repeat prompts are suppressed, scoped to a key such as site or condition.
Rapid dismissal	A prompt answered in under fifteen seconds, or as one of more than five answers from one operator in five minutes (Calculation 3).
Sample gate	The rule that a not-needed rate computed from fewer than twenty answers is not acted on or published.
Signal vocabulary	The constrained list of signal types (nine in the system described) that rules may fire on.
Supersession	The explicit replacement of one controlled document by another, with the retired version kept queryable.

How Energy Compliance Can Help

This paper asks two things of a rules engine: can each rule actually fire, and has anyone measured whether it is worth an operator's attention. If your fleet cannot answer either one from the record, here is where we come in.

Monitoring Truth Assessment

We point our defect battery at your live monitoring stack, rules layer included, and hand you the findings. Every finding comes back with a source, a timestamp, and the question that resolves it. Where we cannot verify something, it comes back marked deferred with the specific question, not softened into a maybe. You keep the findings whether or not you buy anything afterward.

RenewOps

RenewOps is our renewable operations monitoring and analytics platform, with a Data Center module. NORA, the rules engine this paper takes apart, is part of it, defects and all. If the questions in Chapter 8 are the ones you want your own platform answering, that is the conversation to have.

WattOps Compliance

Chapter 6 treats a procedure library as an evidence source, and that is how an auditor treats it too. WattOps Compliance is our NERC compliance evidence platform (wattopscompliance.com). We will run one standard at one facility through WattOps at no cost, so you see your own evidence position on your own records. CIP standards are excluded from that offer.

Audit defense and Data Requests

When something in the record turns into a compliance question, a senior on our team takes it from there. That means audit preparation and defense, RSAW narratives tied to specific evidence, and Data Request responses mapped to deliverables and deduplicated before anything goes out.

Start with Chapter 8. Run the ten questions against your own rules engine this week. If question 1 or question 9 comes back with a bad answer, call me.

Website: energycomplianceinc.com · Email: rsmith@complianceforenergy.com · Phone: 763.438.4427

About the Author

Rob Smith is the Founder and Principal of Energy Compliance, Inc., with more than thirty years in the industry on every side of the system. He began in control-room operations as a Reliability Coordinator, Transmission Operator, and Power System Operator across RTO/ISO and utility control centers. On the regulator side, he was a senior compliance auditor and subject matter expert for NERC Reliability Standards. He audited grid facilities, evaluated mitigation adequacy, participated in risk-based oversight of utility mitigation activities, and supported the development of violation notifications and settlements in FERC-directed enforcement actions. He conducted a regulatory audit in the Sultanate of Oman against the Sultanate's Sector Law and Grid Code.

He founded Energy Compliance, Inc. to deliver senior regulator-side compliance authority without the layered staffing and billable-hour overhead of larger firms.

Education:

- MBA, Energy Management, University of Mary
- MSL, Compliance Law, Fordham Law University
- BAS, Energy Management, Bismarck State College
- BAS, Mechanical Engineering, Metallurgy minor, University of Florida
- BAS, Criminology, St. Petersburg College (State Law Enforcement Certification)
- AAS, Power Plant Technology, Bismarck State College
- AAS, Electrical Transmission System Technology, Bismarck State College

Certifications: NERC Lead Auditor training; previously NERC-certified at the Reliability Coordinator level; 1st Class A Boiler Engineer License, Minnesota.

Auditing teaches one habit that never leaves: before you believe a number, ask what it would look like if it were wrong.

About Energy Compliance, Inc.

Energy Compliance, Inc. is an independent regulatory compliance and advisory firm serving the energy sector, with a focus on NERC, FERC, and RTO/ISO compliance and a particular concentration in the ERCOT and Texas PUCT markets. The firm helps registered entities and prospective registrants through the full compliance lifecycle: registration, program design, evidence development, RSAW production, mitigation, and audit defense.

The approach is research-analyst-first. Every conclusion is tied to an authoritative source, every narrative is backed by evidence, and every deliverable is built to survive regulator scrutiny. A senior practitioner with regulator-side experience oversees every engagement. We do not staff for billable hours. We staff for outcomes. Where automation can replace manual work, we build the automation. Where senior judgment is required, the senior is in the room.

That discipline extends to a portfolio of compliance technology. RenewOps, the reconciliation layer described throughout this series, is one part of it.

Website: energycomplianceinc.com · Email: rsmith@complianceforenergy.com · Phone: 763.438.4427 · Bloomington, Minnesota

Sources and Authorities

This reference carries no regulatory citations by design (see the Disclaimer). Its measurements are first-party observations of one live production instance on August 24–26, 2026.

Energy Compliance, Inc. series references

- Energy Compliance, Inc., EC-WP-1100, *Missing Is Not Zero* (The Measurement Layer), August 2026. Cited for the failure taxonomy and the plausibility failures. https://energycomplianceinc.com/whitepapers/ec-wp-1100-missing-is-not-zero
- Energy Compliance, Inc., EC-WP-1101, *The Board That Deleted Its Own Alarms* (The Alarm Layer), August 2026. https://energycomplianceinc.com/whitepapers/ec-wp-1101-the-board-that-deleted-its-own-alarms
- Energy Compliance, Inc., EC-WP-1102, *What the Incoming Operator Cannot See* (The Operator Layer), August 2026. https://energycomplianceinc.com/whitepapers/ec-wp-1102-what-the-incoming-operator-cannot-see
- Energy Compliance, Inc., EC-WP-1104, *A Machine May Draft It* (The Procedure Layer), August 2026. https://energycomplianceinc.com/whitepapers/ec-wp-1104-a-machine-may-draft-it
- Energy Compliance, Inc., EC-WP-1105, *Six Ways to Compute 99%* (The Contract Layer), August 2026. https://energycomplianceinc.com/whitepapers/ec-wp-1105-six-ways-to-compute-99
- Energy Compliance, Inc., EC-WP-1106, *Displayed Is Not Citable* (The Evidence Layer), August 2026. https://energycomplianceinc.com/whitepapers/ec-wp-1106-displayed-is-not-citable
- Energy Compliance, Inc., EC-WP-1107, *The Reconciliation Layer* (The Platform), August 2026. https://energycomplianceinc.com/whitepapers/ec-wp-1107-the-reconciliation-layer

Technical reference for the worked calculations

- PostgreSQL Global Development Group, PostgreSQL Documentation (current), Section 4.2.7, Aggregate Expressions (FILTER, Calculation 1), and Section 4.2.8, Window Function Calls (RANGE frames with an interval offset, Calculation 3). https://www.postgresql.org/docs/current/sql-expressions.html

© Energy Compliance, Inc. · EC-WP-1103 · Status as of October 17, 2026